

Frontend Angular - Application de Recrutement

Description

Application frontend Angular développée pour le système de recrutement. Cette application moderne et responsive offre une interface utilisateur complète pour la gestion des offres d'emploi et des candidatures.

Fonctionnalités

Page d'accueil

- Hero section avec appel à l'action
- Recherche rapide d'offres d'emploi
- Affichage des offres en vedette
- Statistiques de la plateforme
- Design responsive et moderne

Authentification

- Connexion utilisateur avec JWT
- Inscription de nouveaux utilisateurs
- Gestion des rôles (Admin, RH, Manager, User)
- Validation des formulaires
- Messages d'erreur et de succès

Gestion des offres d'emploi

- Liste des offres avec pagination
- Recherche et filtrage avancés
- Détails des offres d'emploi
- Interface d'administration pour créer/modifier les offres

Interface d'administration

- Tableau de bord pour les RH/Admin

- Gestion des candidats
- Suivi des candidatures
- Statistiques et rapports

Technologies utilisées

- **Angular 18** - Framework frontend
- **TypeScript** - Langage de programmation
- **Bootstrap 5** - Framework CSS
- **SCSS** - Préprocesseur CSS
- **FontAwesome** - Icônes
- **RxJS** - Programmation réactive

Structure du projet

```
src/
├── app/
│   ├── components/      # Composants réutilisables
│   │   ├── header/      # Navigation principale
│   │   └── footer/      # Pied de page
│   ├── pages/           # Pages de l'application
│   │   ├── home/        # Page d'accueil
│   │   ├── login/       # Connexion
│   │   ├── register/    # Inscription
│   │   ├── job-offers/  # Liste des offres
│   │   ├── job-detail/  # Détail d'une offre
│   │   └── admin/       # Interface d'administration
│   ├── services/        # Services Angular
│   │   ├── auth.ts      # Service d'authentification
│   │   ├── job-offer.ts # Service des offres d'emploi
│   │   └── candidate.ts # Service des candidats
│   ├── models/          # Interfaces TypeScript
│   │   └── interfaces.ts # Définitions des types
│   └── app.routes.ts     # Configuration des routes
├── styles.scss           # Styles globaux
└── index.html            # Page principale
```

Installation et démarrage

Prérequis

- Node.js (version 18 ou supérieure)

- npm ou yarn
- Angular CLI

Installation

Installer les dépendances

npm install

Démarrer le serveur de développement

ng serve

L'application sera accessible sur http://localhost:4200

Build de production

Créer un build de production

ng build --prod

Les fichiers seront générés dans le dossier dist/

Configuration

Variables d'environnement

Modifier les URLs de l'API dans les services selon votre environnement :

// Dans les services

private apiUrl = 'http://localhost:8080/api';

Intégration avec le backend

L'application est configurée pour communiquer avec l'API Spring Boot sur le port 8080. Assurez-vous que :

1. Le backend Spring Boot est démarré
2. CORS est configuré pour accepter les requêtes depuis le frontend
3. Les endpoints API correspondent aux services Angular

Fonctionnalités détaillées

Authentification JWT

- Stockage sécurisé du token dans localStorage
- Intercepteur HTTP pour ajouter automatiquement le token
- Gestion de l'expiration du token
- Redirection automatique selon les rôles

Gestion des rôles

- **ROLE_USER** : Candidats (recherche et candidature)
- **ROLE_HR** : Ressources humaines (gestion des offres et candidats)
- **ROLE_ADMIN** : Administrateurs (accès complet)
- **ROLE_MANAGER** : Managers (consultation des candidats)

Interface responsive

- Design adaptatif pour mobile, tablette et desktop
- Navigation optimisée pour tous les écrans
- Composants Bootstrap personnalisés

Validation des formulaires

- Validation côté client avec Angular Reactive Forms
- Messages d'erreur contextuels
- Feedback visuel en temps réel

API Integration

Endpoints utilisés

- `POST /api/auth/signin` - Connexion
- `POST /api/auth/signup` - Inscription
- `GET /api/job-offers/public` - Offres publiques
- `GET /api/job-offers` - Toutes les offres (authentié)
- `POST /api/candidates/apply` - Candidature avec CV
- `GET /api/candidates` - Liste des candidats (RH/Admin)

Format des données

Les interfaces TypeScript définissent la structure des données échangées avec l'API :

```
interface JobOffer {  
  id: number;  
  title: string;  
  description: string;  
  location: string;  
  salary?: number;  
  contractType: string;  
  status: 'ACTIVE' | 'INACTIVE' | 'CLOSED';  
  createdAt: Date;  
}
```

Personnalisation

Thème et couleurs

Les couleurs principales sont définies dans `src/styles.scss` :

```
:root {  
  --primary-color: #007bff;  
  --secondary-color: #6c757d;  
  --success-color: #28a745;  
  --danger-color: #dc3545;  
}
```

Ajout de nouvelles fonctionnalités

1. Créer un nouveau composant : `ng generate component nom-composant`
2. Ajouter la route dans `app.routes.ts`
3. Implémenter la logique métier dans les services
4. Ajouter les styles SCSS personnalisés

Déploiement

Déploiement sur serveur web

```
# Build de production  
ng build --prod
```

```
# Copier le contenu du dossier dist/ sur votre serveur web
# Configurer le serveur pour servir index.html pour toutes les routes
```

Configuration serveur

Pour le routage côté client, configurez votre serveur web pour rediriger toutes les routes vers `index.html`.

Support et maintenance

Logs et debugging

- Utilisation de `console.log` pour le debugging en développement
- Gestion des erreurs avec try/catch dans les services
- Messages d'erreur utilisateur conviviaux

Performance

- Lazy loading des modules (à implémenter pour les grandes applications)
- Optimisation des images et assets
- Minification automatique en production

Évolutions futures

Fonctionnalités à ajouter

- ☐ Chat en temps réel entre RH et candidats
- ☐ Notifications push
- ☐ Système de favoris pour les offres
- ☐ Calendrier d'entretiens intégré
- ☐ Génération de rapports PDF
- ☐ Intégration avec LinkedIn
- ☐ Tests unitaires et e2e

Améliorations techniques

- ☐ Implémentation du lazy loading
 - ☐ Service Worker pour le mode hors ligne
 - ☐ Optimisation SEO avec Angular Universal
 - ☐ Internationalisation (i18n)
-

Développé avec pour une expérience de recrutement moderne et efficace